

Windfall

第5回電王トーナメントPR文章

作成：井本 康宏 作成日：2017/9/吉日

忙しい人のためのキーワード一覧

これさえあれば、1分で全てわかる

- ▶ Windfallは以下の要素で成り立っています
- 評価関数：確率分布型評価値, 非線形評価関数, 「△2八角戦法」対策
- ゲーム木探索：ベイズ的ゲーム木探索, 最良優先探索, 評価値の不確かさを考慮, フレキシブルな探索深さの変更, 「水平線効果」対策
- 学習：Q(λ)学習, 確率分布に拡張, Kullback–Leibler Divergence最小化問題
- 仕様：2駒関係, 横型Redundant Bitboard
- プログラミング：Visual Studio community 2017, Python, C++, Tensorflow
- その他：長い開発期間, 多数の未実装の機能, ぶっつけ本番

わからなかった方は、次項以降をどうぞ

開発者自己紹介

▶ 開発者

- ▶ 井本 康宏

▶ 職業

- ▶ エンジニア

▶ 棋力

- ▶ 測ったことがないのでわかりませんが、すごく弱いです
- ▶ 学生時代に囲碁・将棋部だったなんて言えない

▶ 開発のきっかけ

- ▶ 趣味、好奇心、研究スキル、プログラミングスキルの向上を目的として、当時、話題だったこともあり開発を始めました

ソフトの名前の由来

- ▶ 英語で「棚ぼた」と意味です
- ▶ 「どんな形でもいいので、勝たせてください」という気持ちを表現できる単語を辞書で探しました
 - ▶ これ以上いいのが見つかりませんでした
 - ▶ 「僥倖」や「望外」も考えましたが、「僥倖」は使用済み、「望外」は違うと思ったので止めました
 - ▶ 名案があれば教えてください

開発コンセプト

- ▶ 「せっかく作るのだったら、独創性にあふれるソフトにしたい」
 - ▶ (少なくとも開発を開始した当時の)既存のソフトの抱える多くの理論的な疑問点・問題点の解決
 - ▶ 例えば、どの局面を探索するか
 - ▶ 探索の深さ、消費時間
 - ▶ 枝刈り
 - ▶ 水平線効果
 - ▶ bias-varianceを考慮しない評価値
 - ▶ 理論的に有効性が確かめられた手法を取り入れれば、簡単に強いソフトが作れるはず

技術的な特徴 1 (評価関数)

- ▶ 評価値を確率分布に拡張
 - ▶ 状態数5(敗勢、劣勢、互角、優勢、勝勢)の多項分布で表現
 - ▶ Δ 2八角戦法に代表される評価値は良いが罨が待っているという状態を検出可能なはず
 - ▶ 局面の価値だけでなく、その不確かさを表現
 - ▶ 分布の平均値：従来の評価値
 - ▶ 分布の分散：評価値の不確かさ
 - ▶ どの局面を深く探索するかを自動的な決定が可能
 - ▶ 探索深さの延長だけでなく、短縮が可能
 - ▶ より効率的な時間消費

技術的な特徴 2 (ゲーム木探索)

- ▶ 評価値の不確かさを考慮した探索
 - ▶ 一般に用いられているアルゴリズムは評価値が絶対的に正しいことを前提としている
 - ▶ 機械学習にbias-varianceは付き物
 - ▶ 評価値が1点でも違うと次の一手が別のものになる可能性がある
 - ▶ 学習が難しくなる原因の一つと考えています
 - ▶ 本ソフトは、評価値の大小ではなく、次の一手を変化させる可能性の大小に基づいて探索
 - ▶ 探索は次の一手を決定できれば十分であり、正確な評価値を求める必要はない

技術的な特徴 3 (学習)

- ▶ Floodgateの棋譜をもとに学習
 - ▶ 集める余裕がなかったので、プロ棋士の棋譜はなし
- ▶ $Q(\lambda)$ 学習
 - ▶ 評価が確率分布なので、それに合わせてアルゴリズムを改造・再定義
 - ▶ Kullback-Leibler Divergenceの最小化問題に帰着
- ▶ 他にも最新のディープラーニングの手法を一部導入
 - ▶ Double-Q学習
 - ▶ Experience replay
 - ▶ Neural Fitted Q Iteration

技術的な特徴 4 (その他)

- ▶ コンピュータ将棋の標準的機能はほとんど未実装
 - ▶ 例えば、定跡データベース、詰めルーチン、指し手関係のハッシュなど
 - ▶ 当然、投了も未実装
- ▶ かろうじてあるコンピュータ将棋らしい部分
 - ▶ 特徴量は2駒関係
 - ▶ 横型Redundant Bitboard
- ▶ 実装
 - ▶ C++で書いたモジュールをPythonから呼び出すハイブリッド仕様
 - ▶ TensorflowをC++から呼ぶ余裕がありませんでした
 - ▶ そしてソースコードが複雑怪奇になる

出来上がるもの

- ▶ 「出来上がったもの」の誤植ではない
- ▶ 当初の計画を大幅に超える開発の長期化
 - ▶ いろいろ忙しくて何もしていない期間が結構長い
 - ▶ 論文を調査している期間も長い
 - ▶ 設計の見直し、ソースコードの書き直しもあった
- ▶ 鋭意開発中
 - ▶ 会社がホワイトなおかげで、プライベートでデスマーチができる、やったね！(錯乱)

今年のご目標

- ▶ 無事に大会に参加
 - ▶ 開発状況がギリギリだが、対局を全うしたい
- ▶ 今までと全く異なるアプローチがあることを宣伝
 - ▶ トップ勢は「乾ききった雑巾を絞る」ような開発をされているようですが、こちらは濡れた雑巾であることをアピールしたい
 - ▶ むしろ、絞る作業ができてなくて水がボタボタと垂れているので、絞り方を教わりたい

将来の展望

▶ 棋風の表現

- ▶ 評価値を確率分布にしたことで局面の捉え方について多くの情報を表現できます
- ▶ 攻め将棋、受け将棋などの棋風を確率分布の比較方法の変更で実現できると考えています
- ▶ また、特定の個人と高い一致率を達成することも同様に可能と考えています
 - ▶ 最善手が存在する以上は、実力者であるほど差し手は当然に一致します

▶ 合議制への応用

- ▶ 本ソフトのみで様々な棋風を表現できるはずなので、他のソフトとバランスをとったり、すべて本ソフトということができると思います
- ▶ 合議制を採られる方がいましたら、是非呼んでください

参考文献(1/2)

- ▶ Bonanzaソース完全解析ブログ <http://d.hatena.ne.jp/LS3600/>
- ▶ Chess Programming WIKI <https://chessprogramming.wikispaces.com/>
- ▶ TDLeaf(λ): Combining Temporal Difference Learning with Game-Tree Search
<https://arxiv.org/abs/cs/9901001>
- ▶ Parametric Return Density Estimation for Reinforcement Learning
<https://arxiv.org/abs/1203.3497>
- ▶ Nonparametric Return Distribution Approximation for Reinforcement Learning
<http://www.ms.k.u-tokyo.ac.jp/2010/ICML2010c.pdf>
- ▶ Best Play for Imperfect Players and Game Tree Search; part I – theory
https://www.researchgate.net/publication/2800367_Best_Play_for_Imperfect_Players_and_Game_Tree_Search_part_I_-_theory
- ▶ Best Play for Imperfect Players and Game Tree Search; part II – experiments
https://www.researchgate.net/publication/2752504_Best_Play_for_Imperfect_Players_and_Game_Tree_Search_part_II_-_experiments

参考文献(2/2)

- ▶ A Bayesian approach to relevance in game playing
<http://www.sciencedirect.com/science/article/pii/S0004370297000593>
- ▶ Q(λ) with Off-Policy Corrections <https://arxiv.org/abs/1602.04951>
- ▶ Safe and Efficient Off-Policy Reinforcement Learning <https://arxiv.org/abs/1606.02647>
- ▶ A new Q(λ) with interim forward view and Monte Carlo equivalence
https://www.researchgate.net/publication/263653362_A_new_Q_l_with_interim_forward_view_and_Monte_Carlo_equivalence
- ▶ Deep Reinforcement Learning with Double Q-learning <https://arxiv.org/abs/1509.06461>
- ▶ Playing Atari with Deep Reinforcement Learning
<https://www.cs.toronto.edu/~7Evmnih/docs/dqn.pdf>
- ▶ Neural Fitted Q Iteration - First Experiences with a Data Efficient Neural Reinforcement Learning Method
<https://pdfs.semanticscholar.org/2820/01869bd502c7917db8b32b75593addfbbc68.pdf>